

1 **CLAIMS**

2 1. A method comprising:

3 forming a scale-independent logical model of an application to be
4 implemented by a distributed computer system, the model having multiple
5 components representing logical functions of the application; and

6 converting individual model components into one or more instances ?
7 representative of physical resources that are used to implement the logical
8 functions.

9
10 2. A method as recited in claim 1, wherein each instance represents a
11 corresponding physical resource in a one-to-one relationship.

12
13 3. A method as recited in claim 1, wherein:
14 the distributed computer system comprises multiple computing nodes;
15 the model components comprise a module that is representative of a
16 functional behavior of the application; and
17 an instance of the module corresponds to a node of the distributed computer
18 system.

19
20 4. A method as recited in claim 1, further comprising allocating the
21 physical resources to implement the instances.

22
23 5. A method as recited in claim 1, further comprising allocating the
24 physical resources in real time to implement the instances.
25

1 6. A method as recited in claim 1, further comprising storing the
2 instances in a database.

3
4 7. A method as recited in claim 6, further comprising querying the
5 database to determine a current configuration of the application.

6
7 8. A computer-readable medium storing computer-executable
8 instructions that, when executed on a computer, perform the method of claim 1.

9
10 9. A method comprising:
11 constructing an application for a distributed computer system according to a
12 logical model, the logical model having multiple components representing logical
13 functions of the application;
14 monitoring operation of the application during runtime; and
15 automatically deploying resources of the distributed computer system to the
16 application as operation conditions change.

17
18 10. A method as recited in claim 9, wherein the application comprises
19 an Internet service.

20
21 11. A method as recited in claim 9, wherein the constructing comprises
22 creating one or more instances of each model component, the instances specifying
23 physical resources used to implement the logical functions.

1 12. A method as recited in claim 11, further comprising storing the
2 instances in a database.

3
4 13. A method as recited in claim 12, further comprising querying the
5 database to determine a current configuration of the application.

6
7 14. A method as recited in claim 9, wherein the monitoring comprises
8 receiving events regarding operating conditions of computer nodes in the
9 distributed computer system and evaluating the events against a policy.

10
11 15. A method as recited in claim 9, wherein the distributed computer
12 system comprises a plurality of interconnected computer nodes and the logical
13 model comprises a module that is representative of a behavior of the application,
14 and the deploying comprises:

15 creating one or more instances of each module component of the logical
16 module, the instances specifying physical resources used to implement the
17 behavior of the application; and

18 allocating a node of the distributed computer system for each instance of
19 each module.

20
21 16. A method as recited in claim 9, further comprising tracking the
22 resources that are deployed to the application and available resources that have not
23 yet been deployed.

1 **17.** A computer-readable medium storing computer-executable
2 instructions that, when executed on a computer, perform the method of claim 9.

3
4 **18.** A method comprising:
5 maintaining a logical model of an application for a distributed computer
6 system, the logical model having multiple components representing logical
7 functions of the application;
8 creating one or more instances of each component in the logical model; and
9 allocating resources of the distributed computer system to implement each
10 of the instances.

11
12 **19.** A method as recited in claim 18, wherein the creating comprises
13 producing multiple identical instances of a component in the logical model to
14 accommodate operating conditions for the logical function represented by the
15 component.

16
17 **20.** A method as recited in claim 18, wherein the distributed computer
18 system comprises a plurality of interconnected computer nodes and the logical
19 model comprises a module that is representative of a behavior of the application,
20 the allocating comprising allocating a node for each instance of each module in the
21 logical model.

22
23 **21.** A method as recited in claim 18, further comprising recording the
24 instances in a database.
25

1 **22.** A method as recited in claim 21, further comprising querying the
2 database to determine a current configuration of the application.

3
4 **23.** A method as recited in claim 18, further comprising tracking the
5 resources that are allocated and correlating the resources with the instances for
6 which the resources are allocated.

7
8 **24.** A computer-readable medium storing computer-executable
9 instructions that, when executed on a computer, perform the method of claim 18.

10
11 **25.** A method comprising:
12 maintaining a logical model of an Internet service hosted on a plurality of
13 interconnected computer nodes, the logical model having modules representing
14 logical functions of the Internet service;
15 creating one or more instances of each module in the logical model;
16 allocating a computer node for each corresponding instance; and
17 configuring each computer node to perform the logical functions
18 represented by the module from which the corresponding instance is created.

19
20 **26.** A method as recited in claim 25, wherein the configuring comprises
21 loading software onto the computer node.

22
23 **27.** A method as recited in claim 25, wherein the configuring comprises
24 downloading software from a remote location to the computer node via a network.
25

1 **28.** A method as recited in claim 25, wherein the configuring comprises:
2 initializing the computer node by installing a platform software; and
3 loading a software program that performs the logical functions associated
4 with the instance.

5
6 **29.** A method as recited in claim 25, further comprising tracking the
7 instances that are created in a database.

8
9 **30.** A method as recited in claim 25, further comprising:
10 adding a new instance of a particular module;
11 allocating a new computer node for the new instance; and
12 loading software onto the new computer node that performs the logical
13 functions represented by the particular module.

14
15 **31.** A method as recited in claim 25, further comprising:
16 removing a particular instance of a particular module;
17 deallocating a computer node associated with the particular instance; and
18 returning the computer node to a pool of available computer nodes.

19
20 **32.** A computer-readable medium storing computer-executable
21 instructions that, when executed on a computer, perform the method of claim 25.
22
23
24
25

00000707 102400

1 **33.** A system to deploy an application for a distributed computer system
2 having a plurality of computer nodes, the system comprising:

3 a logical model of the application, the logical model having multiple
4 components representing logical functions of the application; and

5 a core converter to create one or more instances of the model components
6 and allocate computer nodes of the distributed computer system for the instances
7 to implement the logical functions represented by the model components from
8 which the instances are created.

9
10 **34.** A system as recited in claim 33, wherein the core converter
11 comprises:

12 a service running state to track the instances created for the model
13 components; and

14 a resource manager to track the computer nodes available to be allocated.

15
16 **35.** A system as recited in claim 33, wherein the core converter
17 comprises a loader to load software on the computers nodes to implement the
18 logical functions.

19
20 **36.** A system as recited in claim 33, wherein the core runtime converter
21 comprises:

22 a service running state to track the instances created for the model
23 components;

24 a resource manager to track the computer nodes available to be allocated; and
25

1 a loader to load software onto the new computer node, the software being
2 executable on the computer node to implement the logical functions represented
3 by the particular model component.

4
5 **37.** A system as recited in claim 33, further comprising a management
6 policy to monitor operation of the application and to specify when new instances
7 of the model components are to be created or removed based on the operation of
8 the application.

9
10 **38.** A system as recited in claim 37, wherein the management policy
11 listens to events generated by the computer nodes as a way to monitor operation of
12 the application.

13
14 **39.** A model conversion system comprising:
15 a service running state to maintain a logical model of a service application
16 to be implemented by software distributed across a plurality of computer nodes,
17 the logical model having multiple components representing logical functions of
18 the application;

19 a resource manager to allocate computer nodes for the instances; and
20 a loader to load various software onto the computer nodes allocated by the
21 resource manager, the software being executable on the computer nodes to
22 implement the logical functions represented by the model components from which
23 the instances are derived.

09696707 102400

1 **40.** A system as recited in claim 39, wherein the service running state
2 tracks the instances.

3
4 **41.** A system as recited in claim 39, wherein the resource manager
5 tracks whether the computer nodes are allocated or unallocated.

6
7 **42.** A system as recited in claim 39, further comprising a management
8 policy to monitor operation of the application and to specify when new instances
9 of the model components are to be created or removed based on the operation of
10 the application.

11
12 **43.** A system as recited in claim 42, wherein the management policy
13 listens to events generated by the computer nodes as a way to monitor operation of
14 the application.

15
16 **44.** A system as recited in claim 39, further comprising a node loader
17 resident at each of the computer nodes to install the software onto the computer
18 nodes.

19
20 **45.** A system comprising:
21 means for maintaining a scale-independent logical model of a service
22 application to be implemented by software distributed across a plurality of
23 computer nodes, the logical model having multiple components representing
24 logical functions of the application;
25

1 means for creating one or more instances of the model components
2 according to a desired scale of the service application; and

3 means for allocating the computer nodes to associated instances of the
4 model components, the computer nodes being configured to perform the logical
5 functions represented by the components from which the instances are created.

6
7 46. A system as recited in claim 45, further comprising means for
8 tracking the instances.

9
10 47. A system as recited in claim 45, further comprising means for
11 tracking the allocated computer nodes.

12
13 48. A system as recited in claim 45, further comprising means for
14 facilitating policy that specifies when and what instance of a particular model
15 component is created.

16
17 49. A system as recited in claim 45, further comprising means for
18 loading software onto the computer nodes following allocation, the software being
19 executed on the computer nodes to perform the logical functions represented by
20 the components from which the instances are created.

1 **50.** One or more computer-readable media comprising computer-
2 executable instructions that, when executed on one or more processors, direct one
3 or more computing devices to:

4 maintain a logical model of an application to be implemented by software
5 distributed across a plurality of computer nodes, the logical model having multiple
6 components representing logical functions of the application; and

7 convert the model components into one or more instances representative of
8 physical resources used to implement the logical functions.

9
10 **51.** One or more computer-readable media as recited in claim 50, further
11 comprising computer-executable instructions that, when executed on one or more
12 processors, direct one or more computing devices to allocate the physical
13 resources for each of the instances.

14
15 **52.** One or more computer-readable media as recited in claim 50, further
16 comprising computer-executable instructions that, when executed on one or more
17 processors, direct one or more computing devices to track the instances in a
18 database.

19
20 **53.** A computer-readable storage medium storing a data structure, the
21 data structure comprising:

22 a logical model of an application for a distributed computer system, the
23 logical model having at least one module that represents a functional behavior of
24 the application, at least one port that represents a communication access point for
25

0366707 " 102400

1 the module, and at least one wire that represents a logical connection between the
2 port of the module and a port of another module;

3 a first structure to store module information pertaining to one or more
4 module instances of the module that correspond to physical resources used to
5 implement the functional behavior represented by the module;

6 a second structure to store port information pertaining to one or more port
7 instances of the port; and

8 a third structure to store wire information pertaining to one or more wire
9 instances of the wire.

10
11 **54.** A computer-readable storage medium as recited in claim 53,
12 wherein the module information in the first structure is correlated with the port
13 information in the second structure to associate certain ports with certain modules.

14
15 **55.** A computer-readable storage medium as recited in claim 53,
16 wherein the port information in the second structure is correlated with the wire
17 information in the third structure to associate certain wires with certain ports.

18
19 **56.** A computer-readable storage medium as recited in claim 53,
20 wherein the module information includes data fields selected from a group of
21 fields including an identity of the module instance, an identity of the module in the
22 logical model from which the module instance is created, an identity of a physical
23 computer node on which the module instance is instantiated, an identity of a port
24 for the module, and a protocol supported by the module.

25

1 **57.** A computer-readable storage medium as recited in claim 53,
2 wherein the port information includes data fields selected from a group of fields
3 including an identity of the port instance, an identity of the port in the logical
4 model from which the port instance is created, a network address of a physical
5 computer node on which the port instance is instantiated, an identity of a module
6 with which the port is the communication access point, and a protocol supported
7 by the port.

8
9 **58.** A computer-readable storage medium as recited in claim 53,
10 wherein the wire information includes data fields selected from a group of fields
11 including an identity of the wire instance, an identity of the wire in the logical
12 model from which the wire instance is created, an identity of a port with which the
13 wire is coupled, and a protocol supported by the wire.